JOURNAL OF THE EXPERIMENTAL ANALYSIS OF BEHAVIOR 2004, **81,** 297–317 NUMBER 3 (MAY)

A COMPUTATIONAL MODEL OF SELECTION BY CONSEQUENCES

J. J McDowell

EMORY UNIVERSITY

Darwinian selection by consequences was instantiated in a computational model that consisted of a repertoire of behaviors undergoing selection, reproduction, and mutation over many generations. The model in effect created a digital organism that emitted behavior continuously. The behavior of this digital organism was studied in three series of computational experiments that arranged reinforcement according to random-interval (RI) schedules. The quantitative features of the model were varied over wide ranges in these experiments, and many of the qualitative features of the model also were varied. The digital organism consistently showed a hyperbolic relation between response and reinforcement rates, and this hyperbolic description of the data was consistently better than the description provided by other, similar, function forms. In addition, the parameters of the hyperbola varied systematically with the quantitative, and some of the qualitative, properties of the model in ways that were consistent with findings from biological organisms. These results suggest that the material events responsible for an organism's responding on RI schedules are computationally equivalent to Darwinian selection by consequences. They also suggest that the computational model developed here is worth pursuing further as a possible dynamic account of behavior.

Key words: computational modeling, mathematical modeling, selection by consequences, quantitative law of effect, random interval schedules

A fairly successful mathematical *mechanics,* or descriptive account, of operant behavior has been under development in behavior analysis for many years. A mechanics of behavior tells *how* behavioral and environmental variables are related to each other. Part of this mechanics is the well-known quantitative law of effect,

$$R = \frac{kr}{r + r_e}, \qquad (1)$$

where R represents response rate, r represents reinforcement rate, and k and r_e are parameters of the equation (Herrnstein, 1970). The quantitative law of effect tells how behavior is regulated by reinforcement. The hyperbolic form of this expression has been studied extensively, and there is essentially no doubt that the relation between response and reinforcement rates is hyperbolic (McDowell, 1988).

The development of a mathematical *dynamics,* or causal account, of operant behavior has also been pursued, but with less success. A dynamics of behavior tells *why* behavioral and environmental variables are related in the way they are. Different types of dynamic accounts have been proposed, including accounts based on maximization or optimality principles (Rachlin, Battalio, Kagel, & Green, 1981), and accounts based on melioration (Herrnstein, 1982; Vaughan, 1981), to take two prominent examples. These approaches typically use analytic mathematical methods to obtain a mechanical outcome, like Equation 1, from a statement of a dynamic principle or theory. For example, a maximization theory might be stated in terms of a utility function that expresses utility or benefit to the organism as a function of independent variables like reinforcement rate and response cost. The organism is presumed to maximize utility, and hence the maximum of the utility function is taken to be the mechanical prediction of the dynamic account. This maximum is typically an equation that expresses a property of behavior, such as response rate, in terms of independent variables of interest. The various dynamic accounts of operant behavior that have been proposed remain controversial, and none has received wide acceptance.

A different kind of causal principle that has been proposed for operant behavior is selection by consequences (Skinner, 1981). This is a different kind of principle because it does not entail an end state to which behavior is presumed to be directed (like maximum util-

Jesse Dallery, Jordana Richter, and Paul Soto made many helpful comments on an earlier version of this article.

Please direct correspondence to J. J McDowell, Department of Psychology, Emory University, Atlanta, Georgia 30322 (e-mail: psyjjmd@emory.edu).

ity in a maximization account, or equal local rates of reinforcement in melioration), and because it operates at the low level of individual behaviors. A useful dynamic account based on selection by consequences would specify the rules governing the interaction between behavior and its consequences, and these rules would cause some behaviors to be retained over time while permitting others to die out. In this type of dynamic account there is no end state or high-level condition that behavior must satisfy; there are only the low-level rules of selection. Behavior is simply built up by these rules, unconstrained by other conditions. One can conceive of behavior in the typical dynamic account as being pulled to a mechanical outcome by the end-state requirement. In an account based on selection by consequences, on the other hand, one can conceive of behavior as being pushed to a mechanical outcome by rules of selection. Because of these differences, selection by consequences cannot readily be instantiated in analytic mathematical form. However, computational approaches, which are a contemporary alternative to analytic mathematics, can be used to develop a dynamic account based on selection by consequences.

Computational mathematics has been applied, sometimes successfully, to problems in a range of scientific disciplines (Bentley, 2002; Wolfram, 2002a, 2002b), including behavior analysis (e.g., Donahoe, Burgos, & Palmer, 1993; Shimp, 1992). In a computational account, the high-level, or complex, behavior of a system is generated by repeatedly applying simple rules to low-level properties of the system. The high-level behavior of the system is then compared with observation. An important distinguishing feature of a computational account is that the system's high-level behavior cannot be predicted by examining or mathematically manipulating the low-level rules. Instead, the rules must be applied in a computational experiment to discover the complex outcome. Because there is no analytic mathematical connection between the low-level rules and the complex behavior they produce, the latter is sometimes said to be an emergent property of the rules that constitute the computational account.

There have been extensive theoretical and conceptual discussions of selection by consequences in the behavioral literature (e.g., Glenn & Field, 1994; Glenn & Madden, 1995; Hull, Langman, & Glenn, 2001; Skinner, 1984; Smith, 1983). Donahoe and his colleagues (Donahoe et al., 1993; Donahoe & Palmer, 1994; Donahoe, Palmer, & Burgos, 1997) have supplemented this discussion with neural network models that implement a form of selection, namely, selection of neural pathways. These models have not yet been developed extensively, and they typically have been applied to basic conditioning phenomena such as acquisition, extinction, and blocking rather than to higher-order quantitative relations like Equation 1. Moreover, Donahoe and his colleagues have used these connectionist models principally as conceptual tools to illustrate the plausibility of neural mechanisms for conditioning and learning phenomena, rather than as comprehensive mathematical models of those phenomena. This is also a characteristic of much neural network research in cognitive psychology (e.g., Elman et al., 1996).

The purpose of this article is to introduce a computational dynamics of behavior that instantiates Darwinian selection by consequences in a straightforward way. Computational experiments will show that a hyperbolic relation between response and reinforcement rates, namely, Equation 1, is an emergent property of this account.

THE EVOLUTIONARY ALGORITHM

It will be helpful to begin by describing the evolutionary algorithm in general terms. The algorithm begins with a population of behaviors sorted into classes. The population can be conceived of as an organism's behavioral repertoire at a moment of time, t_i. Let the number of individual behaviors falling into each class determine the probability that an instance of that class will be emitted at time, t_i. If the emitted behavior is selected (i.e., reinforced), then the fitness of each behavior in the population must be calculated, where fitness is determined by the similarity of each behavior to the reinforced behavior. Parent behaviors that will produce the next generation are then chosen on the basis of their fitness such that, by and large, fitter individuals are more likely to be chosen as parents. The parent behaviors produce a new generation of behaviors at time, t_{i+1}, and the new generation undergoes a small amount of random mutation. Because relatively fit parents produce the new generation, and because parents tend to produce child behaviors that resemble themselves, the new generation is likely to contain more members of the class of behavior that was reinforced than the old generation, and hence the likelihood that a behavior in that class will be emitted again is increased. Based on the new probabilities for the population of behaviors at time, t_{i+1}, the organism emits an instance of one of the classes of behavior at that moment. The process of selection, reproduction, and mutation is then repeated. If at any time, t_i, an emitted behavior is not reinforced, then parent behaviors may be chosen without regard to fitness. These parents then produce the next generation of behaviors, which undergoes a small amount of random mutation. This general algorithm can be used to create a digital organism that emits a behavior at each moment of time.

IMPLEMENTING THE EVOLUTIONARY ALGORITHM

A working implementation of the evolutionary algorithm consists of five components: (a) a digital organism, (b) a method of calculating fitness, (c) a method of choosing parents for mating based on fitness, (d) a means of reproduction whereby parent behaviors produce child behaviors that resemble themselves, and (e) a method for introducing mutation into the population of behaviors that constitute the organism's repertoire.

The Organism

Let a digital organism's repertoire consist of 100 behaviors, each of which is defined by an integer from 0 through 1023. Evidently, there are 1024 different behaviors that can appear in this repertoire, although only 100 are present at any one moment. Both the size of the repertoire (viz., 100 behaviors) and the range of integers used to define its members are entirely arbitrary. It may be helpful to think of the repertoire, or population of behaviors, as a bar chart consisting of 100 bars, in which the height of each bar defines the behavior. Behavioral classes are obtained by partitioning the possible range of integer values that define the individual behaviors. For example, four behavioral classes might be defined such that the first consists of the integer values from 0 through 255, the second consists of values from 256 through 511, the third consists of values from 512 through 767, and the fourth consists of values from 768 through 1023. In this example, each behavioral class contains 256 possible individual behaviors.

Now suppose 100 behaviors are selected at random from the 1024 possible individual behaviors. Each of these behaviors will fall into one of the four behavioral classes. The proportion of behaviors that fall into a specific class represents the probability that a behavior in that class will be emitted. Evidently, these probabilities sum to one, and in our example, the probabilities will be roughly equal. If another set of 100 behaviors is chosen randomly, the probabilities again will be roughly equal, and so on. Each time a random set of 100 behaviors is chosen, a behavior from one of the four behavioral classes will be emitted, based on their probabilities. In our example, the frequencies of behavioral emissions from the four classes will be approximately equal over time. The number of possible individual behaviors in each class thus defines a base probability of emission for that class. In the example used here, the base probability is 256/1024 or 0.25 for each class. This is an operant level. The base probabilities of course need not be equal. For example, we may define four behavioral classes in such a way that 41 individual behaviors fall into each of two classes, and 471 individual behaviors fall into each of the other two classes, for a total of 1024 possible individual behaviors. In this case the base probabilities of two classes are 41/1024 or 0.04, and the base probabilities of the other two classes are 471/1024 or 0.46. Evidently, the operant levels for the former two classes are much lower than for the latter two classes.

Individual behaviors in the organism's repertoire also have what might be referred to as a genotype. The integer value of each behavior can be considered the phenotypic expression of a 10-digit string of 0s and 1s, which is the behavior's genotype. This string is the binary representation of the integer value. For example, a behavior with an integer value of 235 can be viewed as expressing its genotype of 0011101011, which is the 10-digit binary representation of 235. Ten-digit binary numbers can range from 0000000000 to 1111111111, which translates into a range of 0 to 1023 in base-10 integers. Interestingly, one can view a behavior's genotype as consisting of a single digital chromosome that is made up of 10 genes (the ten characters in the string of 0s and 1s), each of which has two alleles (0 and 1).

Fitness

When a behavior is selected, or reinforced, it is identified as ''fit'' with respect to environmental conditions. The other behaviors in the repertoire can be considered more or less fit depending on how similar they are to the selected behavior. If there are four behavioral classes, the first one consisting of integer values from 0 through 40, and an instance of that class has been selected, one could define a fitness criterion as the middle value of the class, which in this example is 20. The other behaviors in the population are fitter the closer they are to this criterion value. The fitness of each behavior can be defined as the absolute value of the difference between the criterion value and the integer that defines the behavior. Smaller absolute differences correspond to fitter individuals. Fitness defined in this way will be referred to as *midpoint fitness.* Of course, there are other ways to define fitness. For example, a specific behavior in the selected class could be chosen probabilistically as the fitness criterion, using the relative frequency of each behavior in the selected class as the probability that it will be chosen as the criterion. The fitness of each behavior in the repertoire would then be calculated as before, except that the specific behavior chosen from the class, rather than the midpoint of the class, would serve as the fitness criterion. Fitness defined in this way will be referred to as *specific individual fitness.*

Parents

Once a behavior has been selected and the fitness of each behavior in the population has been calculated, parent behaviors must be chosen for mating on the basis of their fitness. Fitter behaviors should be more likely to be chosen for mating than less fit behaviors. The relation between a behavior's fitness and its probability of being chosen as a parent will be referred to as a *parental fitness function.* For example, one could define a parental fitness function such that all behaviors with fitnesses less than or equal to a specific value, *a,* are equally likely to be chosen for mating. Remember that smaller fitness values indicate fitter behaviors. This is a *uniform* parental fitness function, in which the probability density associated with fitness values between 0 and *a* is

$$p(x) = \frac{1}{a} \quad \text{for } 0 \leq x \leq a. \tag{2}$$

For a uniform parental fitness function, behaviors with fitnesses greater than *a* have no chance of being selected for mating. The probability of a parent being at least as fit as some value, *x,* is obtained by integrating Equation 2 from 0 to *x,* which yields the cumulative density function

$$P(x) = \frac{1}{a}x. \tag{3}$$

And finally, for a uniform parental fitness function, the mean fitness of behaviors chosen for mating is

$$\mu = \int_0^a xp(x)\ dx = \int_0^a x\frac{1}{a}\ dx = \frac{1}{2}a. \tag{4}$$

To implement a parental fitness function computationally, it is usually necessary to write it as a probability density function (e.g., Equation 2), and to calculate its cumulative density function and mean (e.g., Equations 3 and 4). In principle, any function form that associates higher probabilities of being chosen for mating with lower (and hence better) fitness values can serve as a parental fitness function. A method for obtaining a probability density function, cumulative density function, and mean for any appropriate function form is described in the Appendix.

Given a parental fitness function, a father behavior can be selected by drawing a fitness value at random from the parental fitness function and then searching the population of behaviors for an individual with that fitness. If a qualified individual is not found, then another fitness value is drawn at random, and the population of behaviors is searched again, and so on, until a qualified father behavior is found. A mother behavior may be obtained in the same way, with the constraint that it must be distinct from the father behavior.

Reproduction

A child behavior can be produced from two parents by building a 10-digit string of 0s and 1s from the parents' genotypes. The digit at each location in the child's string comes from the same location in the digital chromosome of either the father or the mother, with a probability of 0.5 that it comes from a specific parent. Hence the first digit in the string will be either the father's first digit or the mother's first digit; the second digit in the string will be either the father's second digit or the mother's second digit, and so on. The child's phenotype is the base-10 representation of the binary number defined by the string of 0s and 1s. The process of choosing mates that then produce a child behavior continues until the new generation is populated with 100 behaviors. Reproduction that occurs in this way will be referred to as *bitwise* reproduction. Notice that a given parent behavior may be selected for mating more than once, and consequently may have more than one mate and produce more than one child behavior.

Other methods of reproduction, of course, can be implemented. For example, one might slice the parental chromosomes at a random location, combine the left piece of the father's chromosome with the right piece of the mother's, and the right piece of the father's chromosome with the left piece of the mother's, and then take one of these combinations at random as the child. Reproduction that occurs in this way will be referred to as *slicewise* reproduction.

Mutation

Once a new generation of behaviors has been produced, it undergoes random mutation. A specific percentage of individual behaviors selected at random from the population become mutants, that is, their integer values change. This percentage may be less than 1, in which case mutation will not occur in every generation of 100 behaviors.

One way to obtain the integer value of a mutant is to assume that the original integer value of the behavior selected for mutation is the mean of a Gaussian distribution of integer values with a specific standard deviation. The integer value of the mutant is then selected at random from this distribution. This method of mutation will be referred to as *Gaussian* mutation. When implementing Gaussian mutation it is necessary to specify the percentage of individual behaviors that will undergo mutation and the standard deviation of the Gaussian distribution used to obtain the integer value of the mutant.

Another way to obtain the integer value of a mutant is to flip, or change, exactly one of the bits of the behavior selected for mutation. A bit location is selected at random from the behavior's digital chromosome and the bit at that location is flipped from 0 to 1 or 1 to 0. This method of mutation will be referred to as *bit-flip* mutation. When implementing bit-flip mutation it is only necessary to specify the percentage of individual behaviors that will undergo mutation.

Yet another way to obtain the integer value of a mutant is to choose one randomly from the range of possible behaviors (0 through 1023), and simply replace the integer value of the behavior selected for mutation with the randomly selected value. This method of mutation will be referred to as *random individual* mutation. When implementing this mutation method it is only necessary to specify the percentage of behaviors that will undergo mutation.

For the experiments reported in this article, the five components of the evolutionary algorithm were implemented in a computer program that created a digital organism. The digital organism generated behavior according to the rules of the evolutionary algorithm described above and worked on random-interval (RI) schedules in three series of computational experiments. In the first series of experiments (Phase 1), the behavior of the digital organism was studied using different forms for the parental fitness functions. In the second series of experiments (Phase 2), the organism's behavior was studied using different fitness, reproduction, and mutation methods. And in the third series of experiments (Phase 3), the organism's behavior was studied using different rates of mutation. Taken together, these experiments constituted a qualitative and quantitative parametric study of the behavior of the digital organism on RI schedules.

METHOD

Subject

The subject was a digital organism with a repertoire of 100 behaviors defined by integers ranging from 0 through 1023, partitioned into four behavioral classes. Class 1 consisted of the 41 behaviors with integer values from 0 through 40. Class 2 consisted of the 471 behaviors with integer values from 41 through 511. Class 3 consisted of the 41 behaviors with integer values from 512 through 552. Class 4 consisted of the 471 behaviors with integer values from 553 through 1023. The proportion of behaviors falling into a class at a given moment constituted the probability that the organism would emit an instance of the class at that moment. A behavior from one of the four classes was emitted at each moment of time. The baseline probabilities of emission for the four classes were 0.04, 0.46, 0.04, and 0.46, respectively.

Apparatus and Materials

Software was written and experiments were conducted on computers using the Windows® XP Professional operating system. Computers had at least 498-MHz Intel® Pentium® III processors, 184 MB of RAM, and 11 GB of hard disk space. Software was written in VB. NET.

Procedure

Phase 1. Fitness was defined by the midpoint fitness method. Five experiments were conducted using exponential parental fitness functions (Equation A7 in the Appendix) with average parental fitnesses of 10, 20, 40, 100, and 200. Five experiments were conducted using linear parental fitness functions (Equation A4 in the Appendix) with average parental fitnesses of 10, 20, 40, 100, and 200. And five experiments were conducted using uniform parental fitness functions (Equation 2) with average parental fitnesses of 10, 20, 40, 100, and 200. As explained in the Appendix, the exponential, linear, and uniform parental fitness functions are uniquely specified by their means. In all experiments, the bitwise reproduction method was used to generate child behaviors, and the Gaussian mutation method with a standard deviation of 25 was used to produce mutants of 3% of the behaviors in each generation. If the mutant behavior had an integer value greater than 1023, it was wrapped to the low end of the integer-value range. For example, a mutant with an integer value of 1024 had its value changed to 0. Analogously, if the mutant behavior had an integer value less than 0, it was wrapped to the high end of the integer-value range.

In all experiments, Class 1 constituted the target class of behaviors, analogous to the class of lever pressing behaviors in an operant chamber. Class 1 behaviors were selected (i.e., reinforced) on 9, 10, or 11 RI schedules with mean interreinforcement intervals of 1, 2, 3, 5, 8, 10, 18, 25, 68, 112, or 200 time ticks. Before being placed on a new RI schedule, the digital organism's Class 1 behaviors were reinforced on a fixed ratio (FR) 1 schedule until 25 reinforcements were obtained.

For each condition, the computational model operated according to the following 14-step pseudocode:

1. One hundred behaviors with integer values from 0 through 1023 were selected at random, where each integer value was equally likely to be chosen. This constituted the digital organism's initial repertoire.
2. The 100 behaviors were grouped into the classes defined above and the probability that a behavior would be emitted from each class was calculated.
3. Based on the calculated probabilities, a behavior from one of the classes was emitted.
4. If the emitted behavior belonged to Class 1, the RI schedule was consulted to determine if reinforcement was available. If so, the behavior was reinforced and the computation proceeded to Step 5. If reinforcement was not available, or if the emitted behavior did not belong to Class 1, then the computation proceeded to Step 10.
5. The fitness of each behavior in the repertoire was calculated using the midpoint fitness method.
6. A fitness value was selected at random from the parental fitness function, using the method described in the Appendix, and the repertoire was searched for a behavior with that fitness value. If a qualifying behavior was found, then it was des-

ignated the father behavior. If a qualifying behavior was not found, then a new fitness value was selected at random from the parental fitness function, and the repertoire was searched again for a behavior with the new fitness value. This process continued until a qualified father behavior was found. The process was then repeated to find a mother behavior, which was required to be distinct from the father behavior.

7. The mother and father behaviors were mated using the bitwise reproduction method, which generated a child behavior.
8. Steps 6 and 7 were repeated until 100 child behaviors were produced. These behaviors constituted the new generation. Notice that a behavior may have been a parent more than once, and may have had more than one mate.
9. The computation proceeded to Step 13.
10. A father behavior and a distinct mother behavior were chosen at random from the repertoire.
11. The mother and father behaviors were mated using the bitwise reproduction method, which generated a child behavior.
12. Steps 10 and 11 were repeated until 100 child behaviors were generated. These behaviors constituted the new generation. Notice that a behavior may have been a parent more than once, and may have had more than one mate.
13. Three percent of the behaviors in the new generation, selected at random, were replaced by mutants, using the Gaussian mutation method with a standard deviation of 25.
14. Time advanced one tick, and the computation returned to Step 2.

Phase 2. The midpoint fitness, bitwise reproduction, and Gaussian mutation methods used in Phase 1 were designated as standards. In Phase 2 the standard methods were varied singly, and then in combinations, in six pairs of experiments. In two experiments using different forms for the parental fitness function, the specific individual fitness method was used, along with the standard reproduction and mutation methods. In two experiments using different forms for the parental fitness function, the slicewise reproduction method was used, along with the standard fitness and mutation methods. In two experiments using different forms for the parental fitness function, the bit-flip mutation method was used, along with the standard fitness and reproduction methods. And in two experiments using different forms for the parental fitness function, the random individual mutation method was used, along with the standard fitness and reproduction methods. In two additional experiments the specific individual fitness, slicewise reproduction, and random individual mutation methods were used in combination. And finally, in two experiments the midpoint fitness, slicewise reproduction, and bit-flip mutation methods were used in combination.

Class 1 constituted the target class of behaviors, and was reinforced on the 11 RI schedules used in Phase 1. Before being placed on a new RI schedule, the digital organism's Class 1 behaviors were reinforced on an FR 1 schedule until 25 reinforcements were obtained. The computation executed the 14-step pseudocode used in Phase 1 with the appropriate fitness method substituted in Step 5, the appropriate reproduction method substituted in Steps 7 and 11, and the appropriate mutation method substituted in Step 13.

Phase 3. The standard fitness, reproduction, and mutation methods from Phase 1, along with a linear parental fitness function, were used in all Phase 3 experiments. All possible combinations of five mean parental fitness values and five mutation rates were studied in these experiments. The five mean parental fitness values were 10, 20, 40, 100, and 200; the five mutation rates were 1, 5, 10, 20, and 50%. The standard deviation for the Gaussian mutation method was 25 in all cases.

Class 1 constituted the target class of behaviors, and was reinforced on the 11 RI schedules used in Phase 1. Before being placed on a new RI schedule, the digital organism's Class 1 behaviors were reinforced on an FR 1 schedule until 25 reinforcements were obtained. The computation executed the 14-step pseudocode used in Phase 1, with the appropriate mutation rate substituted in Step 13.

In all phases, Class 1 behaviors and reinforcements were recorded in 500-tick blocks.

Table 1

Mean and maximum parental fitnesses, number of generations run per RI schedule (in thousands), parameters of the best-fitting hyperbola, and the proportions of variance accounted for (VAC) by the hyperbola (Hyp), an asymptotic exponential (A Exp), an asymptotic power function (A Power), and a ramp function for Phase 1 experiments. Asterisks mark fits for which there was a statistical indication of nonrandom residuals.

Experiment	Parental fitness		Generations	Hyperbola parameters		Proportion VAC			
	Mean	Max	(000s)	k	r_e	Hyp	A Exp	A Power	Ramp
Exponential parental fitness function									
E160	10	∞	20	480	19	1.00*	0.98*	0.99	0.94*
E66	20	∞	20	413	18	1.00	0.98*	0.98*	0.91*
E67	40	∞	20	320	13	1.00	0.97*	0.98*	0.89*
E68	100	∞	20	208	11	1.00*	0.98*	0.97*	0.88*
E70	200	∞	20	148	13	0.99	0.98*	0.98	0.92*
Linear parental fitness function									
E72	10	30	5	471	19	1.00*	0.97*	0.99	0.91*
E75	20	60	20	331	12	1.00	0.97*	0.96*	0.90*
E76	40	120	20	251	11	1.00	0.96*	0.92*	0.63*
E77	100	300	20	136	7	0.99	0.95	0.92*	0.81
E78	200	600	30	78	7	0.98	0.96	0.93*	0.83*
Uniform parental fitness function									
E73	10	20	20	479	20	0.99*	0.96*	0.99	0.89*
E74	20	40	20	269	9	1.00	0.99*	0.91*	0.92*
E80	40	80	20	192	6	1.00	0.97*	0.83*	0.91*
E163	100	200	20	90	5	0.98	0.94	0.74*	0.86*
E82	200	400	30–45	35	2	0.91	0.91*	0.91	0.80
					Median	1.00	0.97	0.96	0.89

Sessions continued at each RI value until the mean number of emissions of Class 1 behavior per 500-tick block, excluding the first block, was essentially stationary, that is, its standard error was small.

RESULTS

Phase 1

Emission and selection frequencies in Phase 1 were averaged over 10 to 90 500-tick blocks, which represented 5,000 to 45,000 generations of behavior on each RI schedule. Equation 1 was fitted to the data from the 15 Phase 1 experiments by the method of least squares. The results are listed in Table 1, along with the mean and maximum parental fitness for each experiment, and the number of generations run for each RI schedule, excluding the first 500 generations, which were discarded. Selection and emission frequencies were averaged over the number of generations listed in the table. The parameters of the hyperbola and the proportion of variance it accounted for are listed in columns 5, 6, and 7 of the table. Evidently, the hyperbola accounted for essentially all of the variance in the computational data. The median proportion of variance accounted for by the hyperbola across the 15 experiments was 1.00.

Plots of the mean number of target-class emissions per 500-tick block (response rate) versus the mean number of selections per 500-tick block (reinforcement rate) on each RI schedule are shown in Figure 1 for the three experiments using exponential, linear, and uniform parental fitness functions with a mean parental fitness of 40 (Experiments E67, E76, and E80). Averages over fewer blocks produce scatter more typical of the behavior of a biological organism. For the large numbers of generations used in these experiments, however, the data converged on the specific form shown by the smooth curves in the figure, which are plots of the best-fitting hyperbolas.

The randomness of the deviations from the hyperbolic form was tested by the method recommended by Reich (1992). Three tests were conducted: a proportion procedure test-

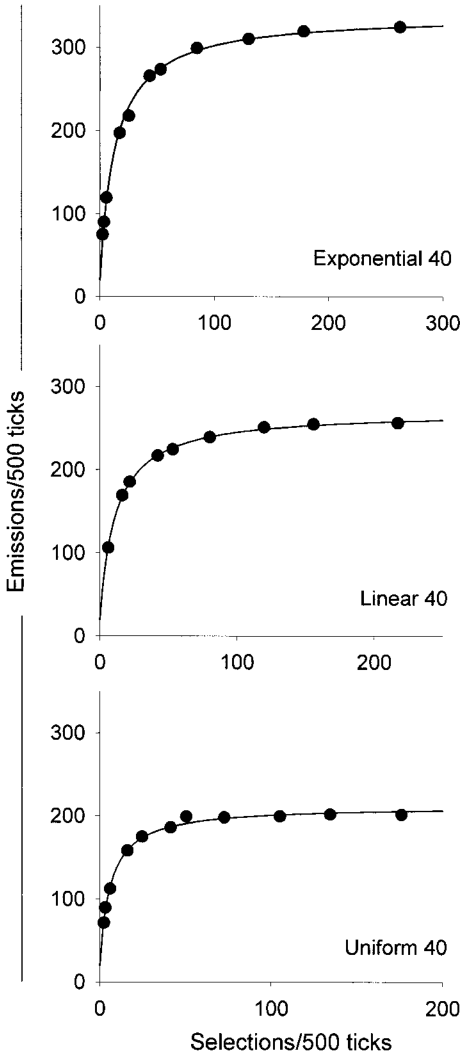

Fig. 1. Plots of the digital organism's response rate versus reinforcement rate (filled circles) on RI schedules using exponential (top graph), linear (middle graph), and uniform (bottom graph) parental fitness functions with means of 40 (Experiments E67, E76, and E80). The smooth curves are the best-fitting hyperbolas.

ed the obtained number of positive residuals for randomness, a runs procedure tested the number of runs of residuals with the same sign for randomness, and a neighborhood correlation procedure tested the correlation of nearby residuals for randomness. If a set of residuals failed one or more of these tests, the proportion of variance accounted for was marked with an asterisk in Table 1. As shown in the table, there was an indication of nonrandom residuals for 4 of the 15 fits of Equation 1. The randomness of the residuals for the entire collection of hyperbolic fits in Table 1 can be tested by calculating the binomial probability of concluding that 4 or more of 15 sets of residuals are not random, given the known probability of falsely concluding that a set of residuals is not random (Type I error). This test is explained in detail in the Appendix. It yielded a binomial probability of 0.16, which is consistent with the conclusion that the residuals for this collection of fits were random.

To test the uniqueness of the hyperbolic description of the computational data, three additional function forms were fitted to the data from each experiment. These were an asymptotic exponential,

$$R = a(1 - e^{-br}),$$

an asymptotic power function,

$$R = a(1 - r^{-b}),$$

and a so-called ramp function,

$$R = \begin{cases} br & 0 \le r \le \dfrac{a}{b} \\ a & r > \dfrac{a}{b} \end{cases}.$$

For all three of these two-parameter equations, R and r represent response and reinforcement rates, respectively, a is the asymptote of the equation (or the upper limit in the case of the ramp), and b governs the rapidity with which the function approaches its asymptote or upper limit. The asymptotic exponential and asymptotic power functions have differential properties that are similar to those of a hyperbola. The ramp function, originally described by Beardsley and McDowell (1992), is piecewise continuous and consists of a line with slope, b, starting at the origin, and a horizontal line, $R = a$, starting at reinforcement rate, a/b. This ramp function provides the simplest two-parameter description of data that increase from zero and then level off. Any more complicated form must provide a better description of the data than a ramp function to be considered seriously.

The proportions of variance accounted for by least squares fits of the asymptotic exponential, the asymptotic power function, and the ramp function are listed in columns 8, 9, and 10 of Table 1. Across all experiments, the asymptotic exponential accounted for a median of 97% of the variance, the asymptotic power function accounted for a median of 96% of the variance, and the ramp function accounted for a median of 89% of the variance, compared to a median of 100% of the variance accounted for by Equation 1. Wilcoxon matched-pairs signed-ranks tests confirmed that the hyperbola accounted for more variance than each of the other function forms ($T = 0$ and $p = 0$ to three decimal places for each of the three comparisons).

The residuals for the fits of the three additional function forms were tested for randomness as described above. If the residuals failed one or more of the three tests, the relevant proportion of variance accounted for was marked with an asterisk in Table 1. As shown in the table, there was a statistical indication of nonrandom residuals for 12 of the 15 fits of the asymptotic exponential, 10 of the 15 fits of the asymptotic power function, and 13 of the 15 fits of the ramp function. The binomial test described in the Appendix confirmed that the sets of residuals for all three additional function forms were not random (the binomial probability was zero to three decimal places for all three sets).

These results show that the hyperbolic form of Equation 1 provides a precise, robust, and unique description of the computational data. The description is precise inasmuch as it accounts for essentially all the variance of the computational data, leaving only random residuals. It is robust inasmuch as it describes the behavior of digital organisms using different parental fitness function forms, and different mean parental fitnesses. And it is unique inasmuch as similar function forms account for significantly less variance, and leave nonrandom residuals.

The values of k and r_e in Table 1 show that these parameters varied systematically with the form of the parental fitness function, and with the mean fitness of each parental fitness function. For a given mean fitness, both k and r_e tended to decrease from the most restrictive (exponential) to the least restrictive (uniform) form of the parental fitness function. A form is more restrictive if it favors fitter parents. In addition, for a given function form, both k and r_e tended to decrease from the most restrictive mean fitness (10) to the least restrictive mean fitness (200). Again, mean parental fitness is more restrictive if it favors fitter parents.

Phase 2

Emission and selection frequencies in Phase 2 were averaged over 10 to 40 500-tick blocks, which represented 5,000 to 20,000 generations of behavior on each RI schedule. Equation 1 was fitted to the data from the 12 Phase 2 experiments by the method of least squares. The results are listed in Table 2, along with the form and mean of the parental fitness function, and the number of generations run for each RI schedule, excluding the first 500 generations, which were discarded. Selection and emission frequencies were averaged over the number of generations listed in the table. In the first eight Phase 2 experiments, a single component of the evolutionary algorithm was changed. For example, the first two experiments in the table were the same as the corresponding experiments in Table 1, except that the specific individual fitness method was used instead of the midpoint fitness method. In the second two experiments listed in Table 2, the slicewise rather than the bitwise reproduction method was used. In the third pair of experiments, the bit-flip rather than the Gaussian mutation method was used. And in the fourth pair of experiments, the random individual rather than the Gaussian mutation method was used. In the remaining four Phase 2 experiments, various combinations of standard and nonstandard fitness, reproduction, and mutation methods were used.

The parameters of the best-fitting hyperbola and the proportion of variance it accounted for are listed in columns 4, 5, and 6 of Table 2. As was the case for the Phase 1 data, the hyperbola accounted for virtually all the variance in the computational data. The randomness of the deviations from the hyperbolic fits was tested as described earlier. There were no statistical indications of nonrandom residuals for the hyperbolic fits.

As before, an asymptotic exponential, an asymptotic power function, and a ramp function were also fitted to the Phase 2 data by the method of least squares. The proportions of variance accounted for by these function forms are listed in columns 7, 8, and 9 of Table 2. The asymptotic exponential accounted for a median of 98% of the variance, the

Table 2

Parental fitness function form and mean, number of generations run per RI schedule (in thousands), parameters of the best-fitting hyperbola, and the proportions of variance accounted for (VAC) by the hyperbola (Hyp), an asymptotic exponential (A Exp), an asymptotic power function (A Power), and a ramp function for Phase 2 experiments. Nonstandard fitness, reproduction, and/or mutation methods are given in the spanner headings. Asterisks mark fits for which there was a statistical indication of nonrandom residuals.

			Hyperbola parameters		Proportion VAC			
Experiment	Parental fitness function form and mean	Generations (000s)	k	r_e	Hyp	A Exp	A Power	Ramp
Specific individual fitness								
E174	Linear 40	20	280	13	0.99	0.96*	0.99	0.71*
E172	Uniform 40	20	227	9	0.99	0.96*	0.98	0.82*
Slicewise reproduction								
E144	Exponential 20	20	417	15	1.00	0.98*	0.98*	0.85*
E175	Uniform 20	20	270	7	0.99	0.98	0.97*	0.91*
Bit-flip mutation								
E143	Exponential 40	20	321	8	1.00	0.98*	0.98	0.90*
E176	Linear 40	20	254	6	1.00	0.97*	0.99	0.81*
Random individual mutation								
E178	Exponential 40	5	339	36	1.00	0.99*	0.93*	0.91*
E179	Linear 40	20	259	31	1.00	0.99*	0.94*	0.68*
Specific individual fitness, slicewise reproduction, random individual mutation								
E188	Exponential 40	5	386	39	1.00	0.99*	0.93*	0.87*
E186	Linear 40	20	281	29	1.00	0.99*	0.96*	0.68*
Midpoint fitness, slicewise reproduction, bit-flip mutation								
E191	Exponential 40	20	334	8	0.99	0.97*	0.88*	0.89*
E189	Linear 40	20	252	6	1.00	0.97*	0.81*	0.81*
				Median	1.00	0.98	0.97	0.84

asymptotic power function accounted for a median of 97% of the variance, and the ramp function accounted for a median of 84% of the variance, compared to a median of 100% of the variance accounted for by the hyperbola. Wilcoxon matched-pairs signed-ranks tests confirmed that the hyperbola accounted for more variance than each of the other function forms ($T = 0$ and $p = 0$ to two decimal places for each of the three comparisons).

The residuals for the fits of the three additional function forms were tested for randomness as described above. If the residuals failed one or more of the three tests, the relevant proportion of variance accounted for was marked with an asterisk in Table 2. There was a statistical indication of nonrandom residuals for 11 of the 12 fits of the asymptotic exponential, 8 of the 12 fits of the asymptotic power function, and all 12 fits of the ramp function. The binomial test described in the Appendix confirmed that the sets of residuals for each of the three additional function forms were not random (the binomial probability was zero to at least three decimal places for each form).

The results from Phase 2 show that the precision, robustness, and uniqueness of the hyperbolic description of the computational data extends to different methods of implementing the fitness, reproduction, and mutation components of the evolutionary algorithm.

Phase 3

Emission and selection frequencies in Phase 3 were averaged over 10 to 40 500-tick blocks, which represented 5,000 to 20,000 generations of behavior on each RI schedule. Equation 1 was fitted to the data from these experiments by the method of least squares. The results are listed in Table 3, along with the mean of the linear parental fitness function, the mutation rate, and the number of generations run for each RI schedule, exclud-

Table 3

Mutation rate, number of generations run per RI schedule (in thousands), parameters of the best-fitting hyperbola, and the proportions of variance accounted for (VAC) by the hyperbola (Hyp), an asymptotic exponential (A Exp), an asymptotic power function (A Power), and a ramp function for Phase 3 experiments. Data for E72, E75, E76, E77, and E78 are from Phase 1. The mean parental fitness of the linear parental fitness function is given in the spanner headings. Asterisks mark fits for which there was a statistical indication of nonrandom residuals.

			Hyperbola parameters		Proportion VAC			
Experiment	Mutation rate (%)	Generations (000s)	k	r_e	Hyp	A Exp	A Power	Ramp
Linear 10 parental fitness function								
E196	1	20	474	7	0.99*	0.93*	0.84*	0.85*
E72	3	20	471	19	1.00*	0.97*	0.99	0.91*
E197	5	5	477	31	0.99*	0.97*	0.97*	0.90*
E198	10	5	472	54	1.00	0.98	0.91*	0.93*
E199	20	5	448	90	1.00	0.99*	0.81*	0.92*
E210	50	5	316	130	1.00	0.99	0.71*	0.97*
Linear 20 parental fitness function								
E192	1	20	344	5	0.99	0.97*	0.71*	0.93*
E75	3	20	331	12	1.00	0.97*	0.96*	0.90*
E193	5	20	338	20	1.00*	0.97*	0.98*	0.87*
E194	10	5	338	37	1.00	0.99*	0.94*	0.91*
E195	20	5	330	66	1.00	0.99*	0.86*	0.94*
E211	50	5	219	83	0.99*	0.99*	0.78*	0.92*
Linear 40 parental fitness function								
E182	1	20	259	4	1.00	0.95*	0.56*	0.89*
E76	3	20	251	11	1.00	0.96*	0.92*	0.63*
E183	5	20	250	16	0.99*	0.96*	0.99	0.89*
E184	10	5	244	29	1.00	0.98*	0.96*	0.81*
E185	20	20	230	50	1.00*	0.99*	0.88*	0.66*
E212	50	5	148	60	1.00	0.99	0.84*	0.85*
Linear 100 parental fitness function								
E200	1	20	134	3	0.96	0.96	0.97	0.90
E77	3	20	136	7	0.99	0.95	0.92*	0.81
E201	5	20	140	11	1.00	0.97*	0.99	0.89*
E202	10	20	137	21	0.99	0.98*	0.97*	0.89*
E203	20	5	116	28	1.00	0.99*	0.94*	0.95*
E209	50	5	78	36	0.99*	0.98*	0.87*	0.93
Linear 200 parental fitness function								
E204	1	20	70	2	0.68	0.53	0.61	0.40
E78	3	20	78	7	0.98	0.96	0.93*	0.83*
E205	5	20	81	10	0.98	0.96	0.97	0.80*
E206	10	20	80	17	0.99	0.98*	0.97*	0.89*
E207	20	20	74	28	1.00	0.99*	0.92*	0.93*
E208	50	5	49	36	0.97	0.97	0.86*	0.85*
				Median	1.00	0.97	0.92	0.89

ing the first 500 generations, which were discarded. The table includes Phase 1 results from E72, E75, E76, E77 and E78. Selection and emission frequencies were averaged over the number of generations listed in the table. The parameters of the best-fitting hyperbola and the proportion of variance it accounted for are listed in columns 4, 5, and 6 of Table 3. Again, the hyperbola accounted for virtually all of the variance in the computational data. The randomness of the deviations from the hyperbolic fits was tested as before, and fits for which there was a statistical indication of nonrandom residuals are marked with asterisks in Table 3. Seven of the 25 Phase 3 fits are so marked. The binomial test described in the Appendix indicated that the residuals for this set of fits were random (the binomial probability was 0.06; Phase 1 data repeated in Table 3 were excluded from the analysis).

An asymptotic exponential, an asymptotic power function, and a ramp function were also fitted to the Phase 3 data by the method of least squares. The proportions of variance accounted for by these function forms are listed in columns 7, 8, and 9 of Table 3. The asymptotic exponential accounted for a median of 97% of the variance, the asymptotic power function accounted for a median of 92% of the variance, and the ramp function accounted for a median of 89% of the variance, compared to a median of 100% of the variance accounted for by the hyperbola. Wilcoxon matched-pairs signed-ranks tests confirmed that the hyperbola accounted for more variance than each of the other function forms ($T = 0$ and $p = 0$ to at least three decimal places for the asymptotic exponential and ramp comparisons; $T = 2$ and $p = 0$ to at least three decimal places for the asymptotic power function comparison; the Phase 1 data repeated in Table 3 were excluded from these analyses).

The residuals for the fits of the three additional function forms were tested for randomness as before. If the residuals failed one or more of the three tests, the relevant proportion of variance accounted for was marked with an asterisk in Table 3. There was a statistical indication of nonrandom residuals for 18 of the 25 fits of the asymptotic exponential, 20 of the 25 fits of the asymptotic power function, and 22 of the 25 fits of the ramp function. The binomial test described in the Appendix confirmed that the sets of residuals for all three additional function forms were not random (the binomial probability was zero to at least three decimal places for all three forms; Phase 1 data repeated in Table 3 were excluded from these analyses).

The results from Phase 3 show that the precision, robustness and uniqueness of the hyperbolic description of the computational data holds over a range of mutation rates, at a variety of mean parental fitnesses.

The values of k and r_e in Table 3 show that these parameters varied systematically with mutation rate. At each mean parental fitness, k decreased moderately and r_e increased markedly as mutation rate increased. The decline in k was gradual over most of the range of mutation rates, and then fell more substantially at the highest rate.

Parameters of the Hyperbola

The effect of mutation rate on k and r_e was different from the effect of mean parental fitness on k and r_e that was observed in Phase 1 and is summarized in Table 1. Changes in each of these variables affected both parameters, but changes in mean parental fitness affected k more strongly than r_e, whereas changes in mutation rate affected r_e more strongly than k. In addition, changes in mean parental fitness caused k and r_e to change in the same direction (both decreased with increasing mean parental fitness), whereas changes in mutation rate caused k and r_e to change in opposite directions.

All of these effects on k and r_e can be seen in Figure 2, which is a plot of the ks against the r_es from Table 3 (filled circles). The plot produces what may be described as a fan with spokes that radiate outward from near the origin (solid lines) and arcs that sweep through the first quadrant from the y- toward the x-axis (dashed lines). Percentages across the top edge of the fan in Figure 2 are the mutation rates associated with each spoke. Numbers along the right edge of the fan are the mean parental fitness values associated with each arc. Traveling outward along a spoke corresponds to decreasing (and hence making more restrictive) the mean parental fitness at a constant mutation rate. As the spokes sweep through the quadrant, mutation rate increases from 1% to 50%. Hence traveling along an arc from left to right corresponds to increasing the mutation rate (and hence diluting the effect of selection) at a constant mean parental fitness. Note that, except for the highest mutation rates (the rightmost spokes), traveling outward along a spoke results in a relatively large increase in k and a relatively small increase in r_e. This is the effect of decreasing the mean parental fitness at a constant mutation rate. Traveling from left to right along an arc results in a relatively small decrease in k and a relatively large increase in r_e, except at the highest mutation rate, where k decreases markedly while r_e continues to increase. This is the effect of increasing the mutation rate at a constant mean parental fitness.

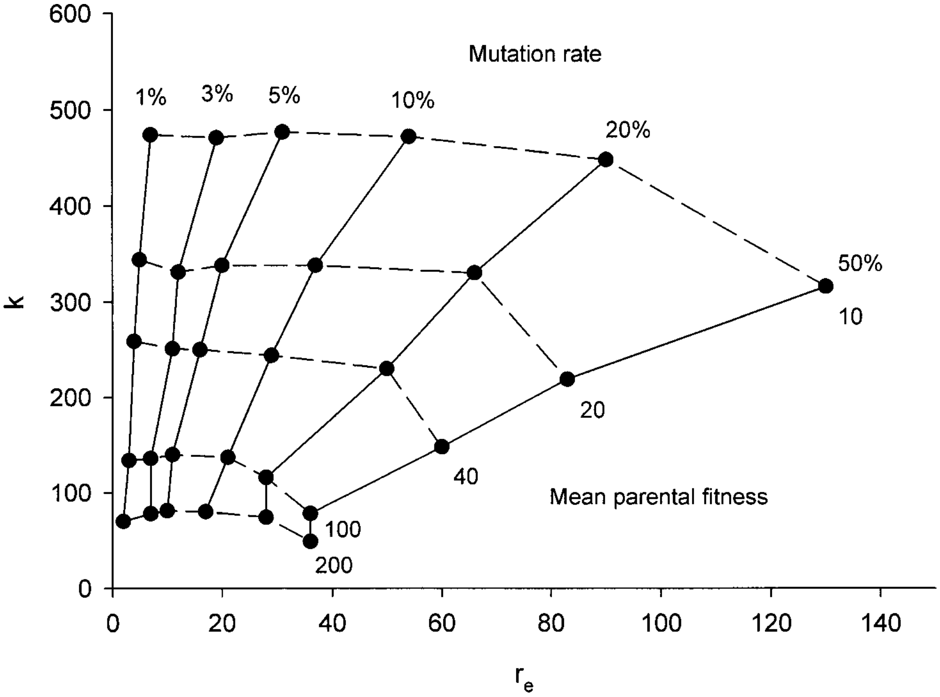

Fig. 2. Plots of k versus r_e from Table 3, which are the parameters produced using a linear parental fitness function with a range of mean fitnesses (listed along the right edge of the parameter fan) and a range of mutation rates (listed along the top edge of the fan). Traveling along a spoke (solid line) outward from near the origin corresponds to a decreasing, and hence more restrictive, mean parental fitness at a constant mutation rate. Traveling along an arc (dashed line) from the y-axis toward the x-axis corresponds to an increasing, and hence more diluting, mutation rate at a constant mean parental fitness.

DISCUSSION

In the absence of reinforcement, the digital organism distributes its behavior according to the baseline probabilities used in its design. Reinforcement can be conceived of as exerting selection pressure against this baseline and pulling the distribution toward the target class of behaviors. Richer RI schedules exert stronger selection pressure than leaner RI schedules. At some point, an RI schedule reaches its maximum effect, in which there is a diffusion-like equilibrium between the RI schedule's tendency to favor the target class of behaviors and the organism's tendency to return to the baseline distribution. Remarkably, as shown by the experiments reported in this article, these equilibrium points trace out a hyperbola. Moreover, the hyperbolic form of the emissions versus selections function in these experiments was precise and unique, did not depend on the form of the parental fitness function or on its mean, did not depend on the specific method of implementing the fitness, reproduction, or mutation components of the evolutionary algorithm, and did not depend on the mutation rate. Hence the hyperbolic form was a robust emergent property of the rules specified by the evolutionary algorithm.

It is clear that the components of the evolutionary algorithm used in these experiments are sufficient to produce a hyperbolic emissions versus selections function. But it is also the case that they are necessary. This is obvious for the fitness method, the parental fitness function, and the reproduction method. In order to implement selection at all, a fitness criterion must be defined, parents must be chosen on the basis of this criterion, and the parents must produce offspring that resemble themselves. But it is important to recognize that mutation is also necessary. In the absence of selection, the digital organism's behavior wanders from class to class and may remain in a particular class for many generations. These sustained response bouts occur when the behaviors in the repertoire are fairly uniform, that is, roughly equal in integer value. New generations produced by randomly selected uniform parents will themselves be fairly uniform, and hence behaviors from the same class will be emitted repeatedly. Mutants, however, occasionally disturb this uniformity and may participate in producing the next generation. When they do, the disturbance may build upon itself and pull the organism's behavior into another class. As a result, the distribution of behavior in the various classes matches the baseline probabilities over time, even though the organism may show sustained bouts of responses from specific classes. If there are no mutants, however, then when the inevitable sustained bouts of responding occur, there is nothing to disturb the uniform behavior and so responding gets stuck in that class for all future generations. This same effect occurs in the presence of selection. If there are no mutants, then the organism eventually allocates either all of its behavior to the target class, regardless of the RI schedule value, or none of its behavior to the target class. If its behavior is uniformly from the target class, then the behavior can never vary enough to break out of the class, even though reinforcement may be infrequent. If the organism's behavior is uniformly from a class other than the target class, then the behavior will never vary enough to come into contact with reinforcement, regardless of how frequently it may be scheduled. Hence mutation is necessary in order to obtain something other than an all-or-nothing allocation of behavior to the target class under the selection pressure of an RI schedule.

Although the specific method of implementing the components of the evolutionary algorithm in these experiments was not critical, many methods of implementation were not tested. One example is asexual reproduction. It may be worthwhile to study this and other methods of implementation, but the critical finding of the present experiments is that the *principle* of Darwinian selection by consequences, rather than any specific implementation of it, is responsible for the hyperbolic form of the emissions versus selections function.

It may seem that the form found in these experiments is somehow inherent in the low-level rules of selection, but this is not likely to be the case. The opposing forces of selection on the one hand, and random reproduction under nonreinforcement and mutation on the other, may generally produce larger emission frequencies at higher selection pressures, but it is not clear that the form of this effect should be precisely hyperbolic. Nor is it clear that the form should remain hyperbolic when the details of the low-level rules are changed. In the absence of an obvious connection between the low-level rules and the hyperbolic form, and given that different versions of the rules produce the same hyperbolic form, it seems unlikely that a connection exists. Instead, the hyperbolic form appears to be a genuine emergent property of the joint action of the component principles of the evolutionary algorithm.

The algorithm developed here bears some similarity to the genetic algorithms studied by Holland (1992, 1995, 1998) and others as models of so-called complex adaptive systems, such as brains, businesses, cities, economies, ecosystems, immune systems, and organisms. The evolutionary algorithm lacks many of the complexities of Holland's genetic algorithms, but it can be expanded into a Holland-like algorithm by adding appropriate structures and functions. This effort might be worth pursuing, especially in attempts to extend the computational model to more complex cases, such as behavior under stimulus control, or social behavior, in which the repertoires of separate organisms interact.

The hyperbolic *form* of the emissions versus selections function did not depend on details of the evolutionary algorithm's low-level rules, but at least some of these details had profound effects on the parameters of the hyperbola. Consider first the form and mean of the parental fitness function. As can be seen from the data in Table 1, both variables had the same effect on k and r_e. As the form or mean of a parental fitness function became more restrictive, that is, as it came to favor fitter parents, both k and r_e increased. Mutation rate also affected k and r_e, but in a different way. As mutation rate increased, k decreased moderately over most of its range, whereas r_e increased markedly. The marked increase in r_e can be understood by recalling that this parameter governs the rapidity with which the hyperbola approaches its asymptote (McDowell, 1988). Because mutation counteracts or dilutes the effect of selection, responding at a given reinforcement rate lies further from the asymptote as the diluting effect of mutation increases. Notice that the form and mean of the parental fitness function directly affect which parents will be selected for mating, whereas the mutation rate operates *after* selection has occurred, and acts to dilute the effect of selection. Notice also that the form and mean of the parental fitness function come into play only when selection occurs, whereas mutation affects every generation of behavior.

Because certain variables, such as reinforcer magnitude, are known to affect the parameters of Equation 1, it is tempting to associate these variables with specific features of the computational model. For example, the cost/benefit ratio for a particular response/reinforcer combination might be represented in the computational model by a parental fitness function with a particular form and mean. A lever press with a given force requirement, say, that is reinforced by water sweetened with a given concentration of sucrose, might be represented by a linear parental fitness function with a specific mean. Changes in the quantitative properties of the response/reinforcer combination, such as the concentration of sucrose, would then correspond to changes in the mean of the parental fitness function. For example, an increase in sucrose concentration would reduce the cost/benefit ratio and hence would be represented by a lower, more restrictive, mean parental fitness. Changes in the qualitative properties of the response/reinforcer combination, such as substituting chain pulling for lever pressing as the target behavior, or food for sweetened water as the reinforcer, might correspond to changes in the form of the parental fitness function.

This feature of the computational model can be examined further by considering the simpler case of a constant response form but a variable reinforcer magnitude. As reinforcer magnitude increases, the cost/benefit ratio of the response/reinforcer combination decreases, and hence mean parental fitness decreases. The data in Table 1 show that as mean parental fitness decreases (due, in our hypothetical example, to increasing reinforcer magnitude), k increases. This is consistent with a substantial body of research that shows that increases in reinforcer magnitude produce increases in k (Dallery, McDowell, & Lancaster, 2000; McDowell & Dallery, 1999; McDowell & Wood, 1984, 1985). Hence these findings support the idea that mean parental fitness in the computational model might represent the cost/benefit ratio of a particular response/reinforcer combination. If so, then the computational model predicts a positive, and approximately linear, relation between k and r_e as the cost/benefit ratio changes. This prediction is shown in Figure 2. Traveling outward along an approximately linear spoke corresponds to decreasing mean parental fitness (and hence cost/benefit ratio) at a constant mutation rate.

Some data bearing on this prediction are provided by Dallery et al. (2000), who studied rats' lever pressing reinforced by 0.00, 0.05, 0.10, 0.20, and 0.32 M concentrations of sucrose in water. Figure 3 shows plots of k versus r_e for Dallery et al.'s 7 rats at the five different concentrations of sucrose, and a plot of k versus r_e pooled across rats (bottom right panel). The vertical error bars represent ± 1 standard error on k; the horizontal error bars represent ± 1 standard error on r_e. The straight lines in each panel were fitted so as to minimize the sum of the proportion of unexplained variance in k and the proportion of unexplained variance in r_e, and were constrained to pass through the origin (i.e., each line was fitted with its slope as the only free parameter). The top number in the bottom right of each panel is the proportion of variance in k accounted for by the straight line; the bottom number is the proportion of variance in r_e accounted for by the straight line. Given the large standard errors in this experiment, the relation between k and r_e for these rats is consistent with the approximately linear relation predicted by the computational model when mean parental fitness is taken to represent the cost/benefit ratio of a particular response/reinforcer combination. Needless to say, additional empirical and computational experimentation will be required to evaluate fully this, and other, features of the computational model.

Herrnstein (1970) originally derived Equation 1 by conceptualizing single-alternative responding as choice between a target alternative and other, extraneous, alternatives,

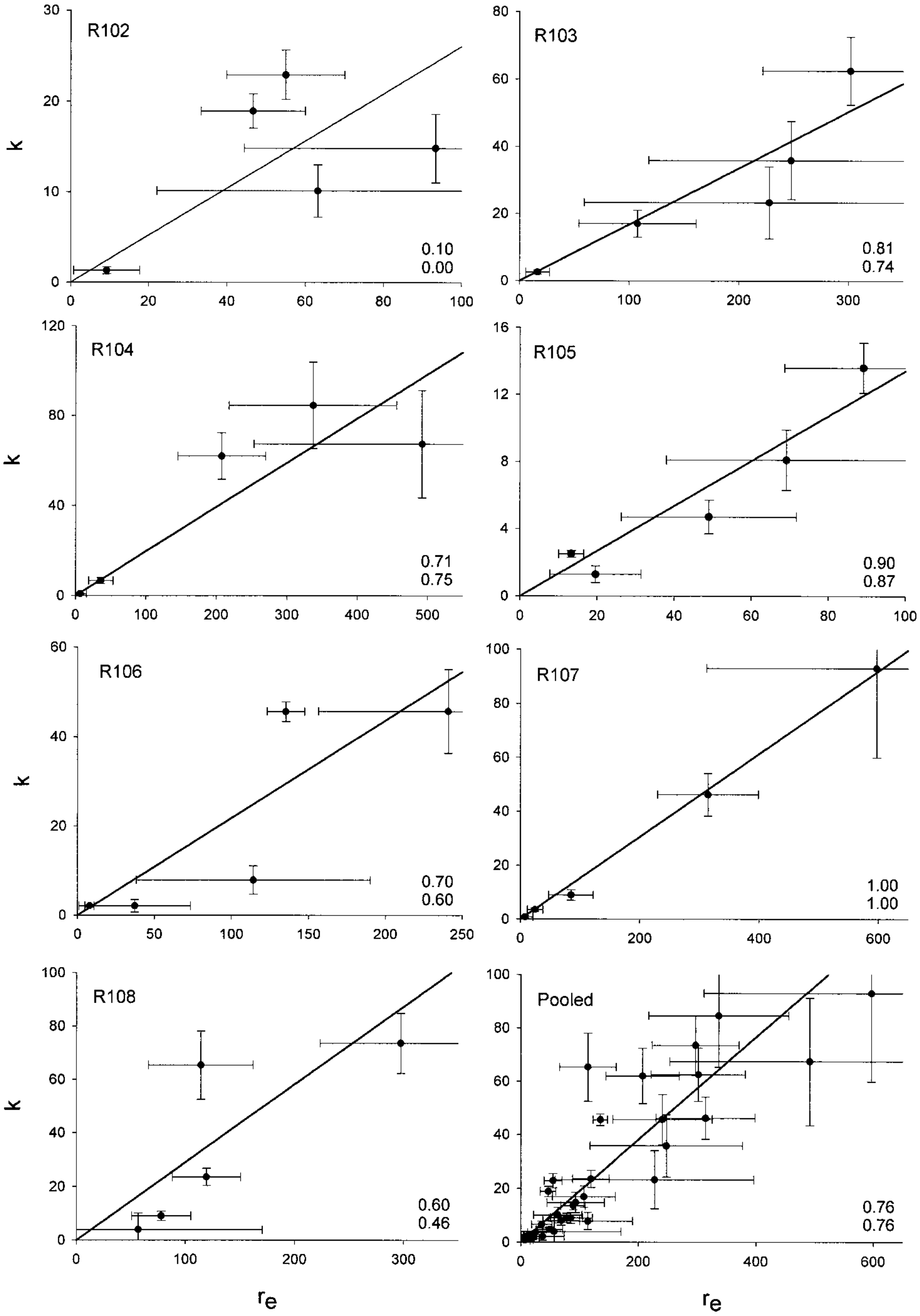

Fig. 3. Plots of *k* versus r_e for Dallery, McDowell, and Lancaster's (2000) 7 rats, whose lever pressing was reinforced by five different concentrations of sucrose in water. In the lower right panel the *k*s and r_es are pooled across rats. The horizontal error bars represent ± 1 standard error on r_e; the vertical error bars represent ± 1 standard error on *k*. The slopes of the straight lines, which were constrained to pass through the origin, were selected so as to minimize the sum of the proportion of unexplained variance in *k* and the proportion of unexplained variance in r_e. The top number in the lower right of each panel is the proportion of variance in *k* accounted for by the line; the bottom number is the proportion of variance in r_e accounted for by the line.

and this choice was presumed to be governed by the matching law. According to Herrnstein's account, the parameters of the hyperbola represented the maximum amount of behavior that is possible to emit, *k*, and the aggregate rate of reinforcement obtained for responding on the extraneous alternatives, r_e. The results of the computational experiments reported in this article provide an alternative to Herrnstein's matching-based derivation of Equation 1. They show that Darwinian selection by consequences is sufficient to produce Equation 1 and that, according to the computational account, the parameters of the equation are related in fairly complicated ways to the low-level rules of selection.

The ability of the computational model to yield Equation 1 as an emergent property means that the material operation of a biological organism responding on RI schedules must be *computationally equivalent* to Darwinian selection by consequences. As a simple example of computational equivalence, consider the process of repeatedly flipping a coin and recording the sequence of resulting heads and tails. This process is computationally equivalent to repeatedly drawing a ball (with replacement) from a bag containing two white and two black balls, and recording the sequence of white and black balls pulled from the bag. And both of these processes are computationally equivalent to selecting an integer at random from the set, {1, 2, 3, . . . , 10}, and recording the sequence of integers-less-than-or-equal-to-5, and integers-greater-than-5. Hence the success of the computational model developed here suggests that whatever specific material events are responsible for an organism's responding on an RI schedule, they must be computationally equivalent to Darwinian selection by consequences. Of course this is strictly the case only if the outcomes of the computational model are completely consistent with behavioral phenomena. The ability of the model to yield Equation 1 as an emergent property, and to accommodate some of the known relations between the equation's parameters and properties of reinforcement, suggest that it is worth pursuing further as a possible dynamic account of behavior.

REFERENCES

Beardsley, S. D., & McDowell, J. J (1992). Application of Herrnstein's hyperbola to time allocation of naturalistic human behavior maintained by naturalistic social reinforcement. *Journal of the Experimental Analysis of Behavior, 57,* 177–185.

Bentley, P. J. (2002). *Digital biology.* New York: Simon & Schuster.

Dallery, J., McDowell, J. J, & Lancaster, J. S. (2000). Falsification of matching theory's account of single-alternative responding: Herrnstein's *k* varies with sucrose concentration. *Journal of the Experimental Analysis of Behavior, 73,* 23–43.

Donahoe, J. W., Burgos, J. E., & Palmer, D. C. (1993). A selectionist approach to reinforcement. *Journal of the Experimental Analysis of Behavior, 60,* 17–40.

Donahoe, J. W., & Palmer, D. C. (1994). *Learning and complex behavior.* Boston: Allyn & Bacon.

Donahoe, J. W., Palmer, D. C., & Burgos, J. E. (1997). The S-R issue: Its status in behavior analysis and in Donahoe and Palmer's *Learning and complex behavior. Journal of the Experimental Analysis of Behavior, 67,* 193–211.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. *Rethinking innateness: A connectionist perspective on development.* Cambridge, MA: MIT Press.

Glenn, S. S., & Field, D. P. (1994). Functions of the environment in behavioral evolution. *The Behavior Analyst, 17,* 241–259.

Glenn, S. S., & Madden, G. J. (1995). Units of interaction, evolution, and replication: Organic and behavioral parallels. *The Behavior Analyst,* 18, 237–251.

Herrnstein, R. J. (1970). On the law of effect. *Journal of the Experimental Analysis of Behavior, 13,* 243–266.

Herrnstein, R. J. (1982). Melioration as behavioral dynamism. In M. L. Commons, R. J. Herrnstein, & H. Rachlin (Eds.), *Quantitative analyses of behavior, Vol. 2: Matching and maximizing accounts.* Cambridge, MA: Ballinger.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence* (2nd ed.). Cambridge, MA: MIT Press.

Holland, J. H. (1995). *Hidden order: How adaptation builds complexity.* Cambridge, MA: Perseus.

Holland, J. H. (1998). *Emergence: From chaos to order.* Cambridge, MA: Perseus.

Hull, D. L., Langman, R. E., & Glenn, S. S. (2001). A general account of selection: Biology, immunology and behavior. *Behavioral and Brain Sciences, 24,* 511–528.

McDowell, J. J (1988). Matching theory in natural human environments. *The Behavior Analyst, 11,* 95–109.

McDowell, J. J, & Dallery, J. (1999). Falsification of matching theory: Changes in the asymptote of Herrnstein's hyperbola as a function of water deprivation. *Journal of the Experimental Analysis of Behavior, 72,* 251–268.

McDowell, J. J, & Wood, H. M. (1984). Confirmation of linear system theory prediction: Changes in Herrnstein's *k* as a function of changes in reinforcer magnitude. *Journal of the Experimental Analysis of Behavior, 41,* 183–192.

McDowell, J. J, & Wood, H. M. (1985). Confirmation of linear system theory prediction: Rate of change of Herrnstein's *k* as a function of response-force requirement. *Journal of the Experimental Analysis of Behavior, 43,* 61–73.

Press, W. H., Flannery, B. P, Teukolsky, S. A., & Vetterling,

W. T. (1989). *Numerical recipes: The art of scientific computing.* Cambridge, England: Cambridge University Press.

Rachlin, H., Battalio, R., Kagel, J., & Green, L. (1981). Maximization theory in behavioral psychology. *Behavioral and Brain Sciences, 4,* 371–417.

Reich, J. G. (1992). *C curve fitting and modeling for scientists and engineers.* New York: McGraw-Hill.

Shimp, C. P. (1992). Computational behavior dynamics: An alternative description of Nevin (1969). *Journal of the Experimental Analysis of Behavior, 57,* 289–299.

Skinner, B. F. (1981). Selection by consequences. *Science, 213,* 501–504.

Skinner, B. F. (1984). Selection by consequences. *Behavioral and Brain Sciences, 7,* 477–510.

Smith, T. L. (1983). Skinner's environmentalism: The analogy with natural selection. *Behaviorism, 11,* 133–153.

Vaughan, W., Jr. (1981). Melioration, matching, and maximization. *Journal of the Experimental Analysis of Behavior, 36,* 141–149.

Wolfram, S. (2002a). *A new kind of science.* Champaign, IL: Wolfram Media.

Wolfram, S. (2002b). *Cellular automata and complexity.* Cambridge, MA: Perseus.

Received January 8, 2003
Final acceptance May 10, 2004

APPENDIX

Calculating Parental Fitness Functions

A parental fitness function must associate higher probabilities of being selected for mating with lower, and hence better, fitness values. Any form with this general property can, in principle, be used as a parental fitness function. In order to implement the parental fitness function computationally, it is usually necessary to write it as a probability density function, and then calculate its cumulative density function and mean. In practice, it is usually easier to obtain the cumulative density function for the desired form first, and then take its first derivative to obtain the probability density function itself, and then finally calculate the mean of the probability density function.

This process can be illustrated by preparing a *linear* parental fitness function. Consider the line,

$$y = -mx + b, \tag{A1}$$

with y-intercept, b, and root (or x-intercept), b/m. When $x = 0$ (i.e., perfect fitness), $y = b$. As x increases (i.e., as fitness decreases), y declines linearly until it reaches 0 at b/m. We are not interested in the line beyond this fitness value because behaviors with fitnesses greater than b/m have no chance of mating.

We can prepare the cumulative density function for this linear form by calculating the integral of the line from 0 to x and then dividing this by the integral of the line from 0 to b/m. The latter integral gives the total area bounded by the line and the x-axis; the former gives that part of the total area up to x. The quotient expresses the area up to x as a proportion of the total area, and hence represents the probability of choosing a fitness of x or better. The numerator and denominator integrals are

$$F_{\text{numerator}}(x) = \int_0^x (-mx + b)\ dx$$

$$= -\frac{m}{2}x^2 + bx \quad \text{and}$$

$$F_{\text{denominator}}(x) = \int_0^{b/m} (-mx + b)\ dx$$

$$= \frac{b^2}{2m},$$

and their quotient is

$$P(x) = -\frac{m^2}{b^2}x^2 + \frac{2m}{b}x, \tag{A2}$$

where $P(x)$ denotes the probability of choosing a fitness of x or better, given the linear function form.

Equation A2 is the cumulative density function for the linear form, Equation A1. The first derivative of Equation A2 gives the linear density function itself:

$$p(x) = \frac{d}{dx}P(x) = -\frac{2m^2}{b^2}x + \frac{2m}{b}, \tag{A3}$$

where $p(x)$ is the probability density associated with a fitness of x. All three equations, A1, A2, and A3, are of interest only on the interval,

$$0 \leq x \leq \frac{b}{m},$$

because behaviors with fitnesses greater than b/m have no chance of mating.

Before calculating the mean of Equation A3, it will be helpful to simplify Equations A2 and A3 by noticing that neither equation depends on the individual values of m and b, but only on their ratio. If we let

$$\frac{m}{b} = a,$$

then Equation A3, the linear probability density function, becomes

$$p(x) = -2a^2x + 2a, \qquad \text{(A4)}$$

and Equation A2, the cumulative density function, becomes

$$P(x) = -a^2x^2 + 2ax \qquad \text{(A5)}$$

for

$$0 \leq x \leq \frac{1}{a}.$$

The mean fitness specified by Equation A4 must be

$$\begin{aligned}\mu &= \int_0^{1/a} x\,p(x)\;dx \\ &= \int_0^{1/a} (-2a^2x^2 + 2ax)\;dx = \frac{1}{3a}. \qquad \text{(A6)}\end{aligned}$$

Equations A4, A5, and A6 are the three expressions that are needed to use a linear parental fitness function in the computational model. The procedure used to obtain these expressions can be applied to any function form that meets the general requirement stated earlier, namely, that higher probabilities of being selected for mating are associated with lower, and hence better, fitness values.

In addition to the uniform parental fitness function developed in the text, and the linear parental fitness function developed here, an exponential parental fitness function may be of interest. It is based on the exponential,

$$y = ae^{-ax},$$

which is already a probability density function,

$$p(x) = ae^{-ax}, \qquad \text{(A7)}$$

defined on the interval,

$$0 < x < \infty,$$

with cumulative density,

$$P(x) = 1 - e^{-ax}, \qquad \text{(A8)}$$

and mean,

$$\mu = \frac{1}{a}. \qquad \text{(A9)}$$

It is important to note that the exponential (Equation A7), linear (Equation A4) and uniform (Equation 2, developed in the text) parental fitness functions depend only on their means. Hence, by specifying an average parental fitness, one specifies a unique parental fitness function, at least for these three forms.

Drawing a Fitness Value at Random

In order to implement the evolutionary algorithm described in this paper it is necessary to draw fitness values at random from a parental fitness function. This can be accomplished by using the exponential, linear, or uniform cumulative density function, Equation A8, A5, or 3. Notice that cumulative probability density, or probability, is a number from 0 to 1. Methods for selecting a random decimal from 0 to 1 are readily available. Linear congruential random number generators like the one described by Press, Flannery, Teukolsky, and Vetterling (1989) are especially popular. Once a random decimal is chosen, it is set equal to the right-hand side of the cumulative density function, which is then solved for fitness value, x. This fitness value then represents a random sample from the parental fitness function.

If we let $u(0, 1)$ represent a random decimal between 0 and 1, then we can use the three cumulative density functions to write expressions for obtaining a random fitness value from a parental fitness function. For the exponential parental fitness function, a random fitness value can be obtained from

$$x = -\mu \ln[1 - u(0, 1)], \qquad \text{(A10)}$$

which is Equation A8 with $u(0, 1)$ substituted for $P(x)$, and a expressed in terms of μ (from Equation A9). The resulting expression is then solved for x to obtain Equation A10.

A random fitness value from the linear parental fitness function can be obtained from

$$x = 3\mu[1 - \sqrt{1 - u(0, 1)}], \qquad \text{(A11)}$$

which is Equation A5 with $u(0, 1)$ substituted for $P(x)$, and a expressed in terms of μ (from Equation A6). The resulting expression is then solved for the root that lies on the interval $(0, 1/a)$ to obtain Equation A11.

A random fitness value from the uniform

parental fitness function can be obtained from

$$x = 2\mu u(0, 1), \qquad \text{(A12)}$$

which is Equation 3 with $u(0, 1)$ substituted for $P(x)$, and a expressed in terms of μ (from Equation 4). The resulting expression is then solved for x to obtain Equation A12.

Binomial Test of Residuals for a Collection of Fits

In the application of Reich's (1992) method described in this paper, three tests for randomness, each with an alpha level of 0.05, were conducted on a given set of residuals. If a set of residuals failed one or more tests, the set was said to show nonrandomness. Under these circumstances, the probability of incorrectly asserting that a set of residuals was not random (Type I error) was not the nominal alpha level of an individual test, but the cumulative alpha level,

$$1 - (1 - 0.05)(1 - 0.05)(1 - 0.05)$$
$$= 0.1426,$$

which is about three times larger than the nominal alpha level. One can compensate for this large cumulative error rate by considering a collection of fits to which Reich's (1992) method has been applied (for example, all the fits in Table 1). Given a hypothetical collection of fits, all with random residuals, one would expect to falsely conclude that about 14% of the sets of residuals were not random. One can compare an obtained collection of fits to this hypothetical collection by calculating the binomial probability of concluding that x or more sets of residuals are not random (where x is the observed number of sets determined to be nonrandom by Reich's method) in a collection of n fits, all with random residuals, where the probability of falsely concluding that a set of residuals is not random is 0.1426. If this binomial probability is less than or equal to 0.05, then x is too large to be consistent with the conclusion that the residuals for the entire collection of fits are random. But if the binomial probability is greater than 0.05, then x indications of nonrandomness do not contradict the conclusion that the residuals are random for the entire collection of fits.